



普通高等教育“十一五”国家级规划教材

教育部“高等学校教学质量与教学改革工程”立项项目

鲁 斌 李 莉 编著

网络程序设计 与开发

计算机科学与技术专业实践系列教材

清华大学出版社





普通高等教育“十一五”国家级规划教材

计算机科学与技术专业实践系列教材

网络程序设计 与开发

鲁 斌 李 莉 编著

清华大学出版社

北京

内 容 简 介

本书是关于网络程序设计方面一本综合、全面、实用的教材,涵盖了现有各种主要的网络程序设计技术,内容包括网络编程基本知识、Windows 套接字基础、MFC Winsock 类编程、WinInet 编程、不同 I/O 模式下的处理方法、HTTP 高级编程、Winsock 控件编程以及基于 ASP.NET 的 Web 编程技术等。通过本书的学习能够使读者掌握目前最流行的 Windows C/S 模式和 B/S 模式网络应用程序的开发技术,使其可以胜任任何复杂程序的设计与开发。本书可用作高等学校网络工程及其相关专业高年级本科生和研究生的教材或教学参考书,也可供其他技术开发人员参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

网络程序设计与开发 / 鲁斌,李莉编著. —北京:清华大学出版社,2010.9

(计算机科学与技术专业实践系列教材)

ISBN 978-7-302-22918-6

I. ①网… II. ①鲁… ②李… III. ①主页制作—程序设计—高等学校—教材

IV. ①TP393.092

中国版本图书馆 CIP 数据核字(2010)第 101159 号

责任编辑:汪汉友

责任校对:梁毅

责任印制:王秀菊

出版发行:清华大学出版社

<http://www.tup.com.cn>

社总机:010-62770175

投稿与读者服务:010-62795954, jsjic@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

地址:北京清华大学学研大厦 A 座

邮编:100084

邮购:010-62786544

印装者:北京鑫海金澳胶印有限公司

经销:全国新华书店

开本:185×260

印张:24.5

字数:588 千字

版次:2010 年 9 月第 1 版

印次:2010 年 9 月第 1 次印刷

印数:1~3000

定价:36.00 元

产品编号:036685-01

普通高等教育“十一五”国家级规划教材
计算机科学与技术专业实践系列教材

编 委 会

主 任：王志英

副 主 任：汤志忠

编 委 委 员：陈向群 樊晓桢 邝 坚
孙吉贵 吴 跃 张 莉

前 言

计算机网络的飞速发展和网络新技术的不断涌现,对网络软件开发人员的程序设计水平和动手实践能力提出了很高的要求。网络软件程序设计具有很强的实用性,是对网络知识和软件知识的综合应用,能够很好地引导网络软件开发人员掌握最新的网络程序设计技术,培养学习的主动性、协作精神和创新能力。然而,在实际教学 and 实践中我们却发现,现有各类教材和教学参考书对于主要的网络程序设计技术讲述不够全面,更不用说适合教学了。这些书往往只是对某些编程技术进行一般性介绍,或只是阐述一些精深的理论,或者只有一些简单的实例,都没有对读者最为关心的技术层次进行系统、全面的探讨。

本书是一部综合、全面、实用的教材,反映了作者多年来的教学思路和经验。与已出版的同类书籍相比,本书更加贴近教学,具有内容全面、重点突出、层次分明、学以致用、配套齐全等特点和教学参考价值,具体体现在:

(1) 贴近教学。目前已出版的各类教材和参考书对于主要的网络程序设计技术讲述不够全面,更谈不上适合教学的内容安排了。本书正是针对此问题而撰写的。

(2) 内容全面。本书的内容覆盖了程序员必须掌握的各个方面:从网络基本知识到 Winsock 的介绍,从基于 Winsock 的主要网络编程技术到基于 ASP.NET 的 Web 编程技术,应有尽有。

(3) 重点突出。网络编程技术和工具种类繁多,不可能面面俱到。本书精心选取了目前主流的网络编程技术进行深入而透彻的分析,从而使得程序员完全可以胜任任何复杂程序设计任务的开发要求。

(4) 层次分明。本书是对作者长期教学与实践经验的总结,在知识点的逻辑结构上提出“层次化”教学理念,将各种主要的网络编程技术按照由底向上,先基础后高级的顺序组织章节,从而使读者不仅“知其然”,更要“知其所以然”,由浅入深,循序渐进,以适合教学的顺序全面地介绍了网络编程的理论、技术和方法。

(5) 学以致用。本书特别强调知识与能力结合,理论与实用并重,各章都有典型的实例讲解,力图培养读者扎实的动手实践能力,使读者能够深入地运用所学技术编制各种网络应用程序。

(6) 配套齐全。本书配套学习素材中收录了书中列举的全部实例程序源代码,不做任何修改即可编译运行,极大地方便了读者学习和使用;同时,为便于教学,还提供了精心制作的多媒体课件供教师使用。

读完本书并且亲自动手实践书中的程序案例之后,读者能够感受到:

(1) 系统而全面地掌握面向 C/S 架构和 B/S 架构的各种主要的网络编程技术;

(2) 层次化的章节组织、通俗易懂的文字描述使得读者学习网络编程技术时不再枯燥难懂,而变得简单易学;

(3) 典型实例的分析和详细的实现过程使得读者运用网络编程技术时得心应手。

本书在内容的安排上,首先简单明了地介绍了网络编程所涉及的基本知识和分类方法,

接着分两个部分展开叙述。第一部分涵盖第 2 章至第 7 章,系统地介绍了基于 Windows Socket 编程接口的各种网络编程技术,内容包括 Windows 套接字基础,MFC Winsock 类编程,WinInet 编程,不同 I/O 模式下的处理方法,HTTP 高级编程,以及 Winsock 控件编程技术等;第二部分涵盖第 8 章至第 11 章,以 ASP.NET 为基础介绍了 Web 编程技术,内容包括 ASP.NET 基础知识,ASP.NET 常用控件与 Page 类,数据访问方法,以及网络购物商城案例等。上述内容安排能够有效地指导读者如何利用现有的主流开发技术进行最流行的 Windows C/S 模式和 B/S 模式的网络应用程序的开发。

本书由鲁斌主编,并负责前 7 章的撰写工作;后 4 章由李莉编写,鲁斌负责统稿。同时,特别感谢程晓荣教授审阅了全书并对本书提出了宝贵建议。另外,研究生孙智慧、宋伟在本书撰写之初做了大量的素材搜集工作,在此一并表示感谢。

由于编者水平有限,错误之处在所难免,恳请广大读者批评指正。

本书可用作高等学校网络工程及其相关专业高年级本科生和研究生的教材或教学参考书,也可供其他技术开发人员参考。

作 者
2010 年 8 月

目 录

第 1 章 绪论	1
1.1 网络编程基础	1
1.1.1 网络应用程序所处的地位	1
1.1.2 网间进程通信面临的问题	1
1.1.3 网间进程的标识	2
1.2 基本网络协议	4
1.2.1 一般特点	4
1.2.2 用户数据报协议	5
1.2.3 传输控制协议	5
1.3 客户机/服务器模型	6
1.3.1 模型的重要性	6
1.3.2 容易混淆的术语	7
1.3.3 模型工作过程与特点	7
1.3.4 C/S 交互方式	9
1.4 网络编程分类	10
1.4.1 基于 TCP/IP 协议栈的网络编程	10
1.4.2 基于 Web 应用的网络编程	10
1.4.3 基于 .NET 的 Web Services 网络编程	10
1.4.4 .NET Remoting 技术	12
1.4.5 电话应用编程接口(TAPI)	13
1.4.6 信报应用编程接口(MAPI)	14
1.4.7 Internet 服务器应用编程接口	14
1.5 本章小结	15
1.6 习题	15
第 2 章 Windows 套接字	17
2.1 Socket 及其来源	17
2.2 套接字的类型	17
2.2.1 流式套接字	18
2.2.2 数据报套接字	18
2.2.3 原始套接字	18
2.3 套接字规范	18
2.3.1 Berkeley 套接字规范	18
2.3.2 Windows 套接字规范	18
2.3.3 Windows 套接字和 Berkeley 套接字的比较	19

2.4	几种典型的 Windows 套接字规范	22
2.4.1	Winsock 1.0	22
2.4.2	Winsock 1.1	22
2.4.3	Winsock 2.0	23
2.5	Winsock 编程的一般模式	23
2.5.1	面向连接的编程模型	23
2.5.2	无连接编程模型	24
2.5.3	几个基本概念	24
2.6	基本操作函数	26
2.6.1	Winsock 的启动和终止	26
2.6.2	Winsock 的创建、绑定与关闭	28
2.6.3	通信连接的建立	34
2.6.4	数据的传输	37
2.6.5	Winsock 的错误处理函数	44
2.6.6	Winsock 的辅助函数	46
2.6.7	Winsock 的信息查询函数	53
2.7	Winsock 2 的新特性	59
2.7.1	对多协议的支持	59
2.7.2	对 I/O 与事件对象的重叠支持	59
2.7.3	套接口组	60
2.7.4	服务质量	61
2.8	Winsock 2 新增函数一览	62
2.9	网络应用程序的运行环境	63
2.10	本章小结	64
2.11	习题	65
第 3 章	MFC Winsock 类编程	66
3.1	MFC 简介	66
3.1.1	MFC 编程框架	66
3.1.2	MFC 对象和 Windows 对象的关系	67
3.1.3	消息映射的实现	68
3.1.4	MFC 应用程序的执行过程	71
3.2	CAsyncSocket 类	72
3.2.1	基本编程模型	72
3.2.2	创建 CAsyncSocket 类对象	74
3.2.3	CAsyncSocket 类可以接收并处理的消息事件	75
3.2.4	连接的请求与接受	76
3.2.5	数据的发送与接收	79
3.2.6	关闭套接字	80
3.2.7	其他成员函数	81

3.3	基于 CAsyncSocket 类的单点聊天程序	82
3.3.1	实现目标	82
3.3.2	创建客户机程序	82
3.3.3	客户机程序的类与消息驱动	90
3.3.4	客户机程序主要功能的代码和分析	91
3.3.5	创建服务器程序	99
3.3.6	服务器程序的流程和消息驱动	101
3.3.7	点对点聊天的服务器程序主要功能的代码和分析	101
3.4	CSocket 类	109
3.4.1	基本编程模型	110
3.4.2	创建 CSocket 类对象	111
3.4.3	连接的建立	112
3.4.4	数据的收发	112
3.4.5	关闭套接字和清除相关对象	112
3.5	基于 CSocket 类的多点聊天程序	112
3.5.1	聊天室程序的功能	112
3.5.2	创建聊天室的服务器程序	113
3.5.3	聊天室服务器程序的主要实现代码和分析	116
3.5.4	创建聊天室的客户机程序	127
3.5.5	聊天室客户机程序的主要实现代码和分析	129
3.6	本章小结	139
3.7	习题	139
第 4 章	WinInet 编程	141
4.1	WinInet API 的一般化问题	141
4.1.1	HINTERNET 句柄	141
4.1.2	WinInet 中错误的处理	142
4.1.3	WinInet 函数中的缓冲区参数	142
4.1.4	WinInet 函数的异步操作模式	142
4.2	基本 WinInet 函数	143
4.2.1	打开一个 WinInet 会话	143
4.2.2	与服务器建立连接	144
4.2.3	回调函数	145
4.2.4	通用文件操作	147
4.2.5	从 Internet 文件读数据	149
4.2.6	向 Internet 文件写数据	150
4.2.7	移动 Internet 文件指针	150
4.2.8	数据可用性查询	151
4.2.9	其他基本函数	151
4.3	FTP 客户机 WinInet 函数	152

4.3.1	一般步骤	152
4.3.2	查询 FTP 服务器上的文件和目录	152
4.3.3	查询或改变 FTP 服务器的当前目录	155
4.3.4	在 FTP 服务器上创建目录	157
4.3.5	在 FTP 服务器上删除目录	158
4.3.6	从 FTP 服务器中下载文件	159
4.3.7	向 FTP 服务器中上传文件	162
4.3.8	从 FTP 服务器中删除文件	163
4.3.9	重命名 FTP 服务器上的文件或目录	164
4.4	MFC WinInet 类	165
4.4.1	概述	165
4.4.2	MFC WinInet 所包含的类	166
4.4.3	基本编程模型	167
4.4.4	WinInet 类简介	170
4.5	基于 WinInet 类的 FTP 客户机编程实例	179
4.5.1	程序要实现的功能	180
4.5.2	创建应用程序的过程	181
4.6	本章小结	192
4.7	习题	192
第 5 章	不同 I/O 模式下的处理方法	193
5.1	阻塞模式与非阻塞模式	193
5.1.1	阻塞模式及其优缺点	193
5.1.2	非阻塞模式及其优缺点	193
5.2	多进程与多线程	194
5.2.1	什么是多进程	194
5.2.2	什么是多线程	195
5.2.3	多进程和多线程的关系	195
5.2.4	网络编程采用多线程机制的重要性	196
5.3	I/O 阻塞模式的多线程网络编程方法	197
5.3.1	MFC 支持的两种线程	197
5.3.2	创建 MFC 的工作线程	198
5.3.3	创建并启动用户界面线程	201
5.3.4	终止线程	204
5.3.5	FTP 客户机多线程编程实例	206
5.4	I/O 非阻塞模式的异步处理模型	215
5.4.1	select 模型	215
5.4.2	WSAAsyncSelect 异步 I/O 模型	218
5.4.3	WSAEventSelect 事件选择模型	222
5.4.4	其他模型	227

5.5	本章小结	228
5.6	习题	228
第 6 章	HTTP 高级编程	230
6.1	HTTP	230
6.1.1	HTTP 的基本概念	230
6.1.2	HTTP 的应用	231
6.2	CHtmlView 类基本知识	232
6.2.1	CHtmlView 类与 WebBrowser 控件	232
6.2.2	CHtmlView 类的成员函数	233
6.3	网络浏览器实例	239
6.3.1	程序实现的基本功能	239
6.3.2	程序实现的一般步骤	240
6.4	本章小结	245
6.5	习题	246
第 7 章	Winsock 控件编程	247
7.1	Winsock 控件的基本操作	247
7.1.1	Winsock 控件的数据传输	247
7.1.2	Winsock 控件的基本属性	247
7.1.3	Winsock 控件的基本方法	248
7.1.4	Winsock 控件的基本事件	248
7.2	电子邮件系统基本原理	249
7.2.1	邮件传输原理	250
7.2.2	电子邮件的地址	250
7.2.3	常用的几种电子邮件协议	250
7.3	简单电子邮件系统的实现过程	251
7.3.1	邮件服务器的实现	251
7.3.2	邮件客户端的实现	258
7.4	本章小结	265
7.5	习题	266
第 8 章	ASP.NET 基础知识	267
8.1	ASP.NET 简介	267
8.1.1	早期的 Web 开发	267
8.1.2	传统的 ASP	269
8.1.3	ASP.NET	269
8.2	ASP.NET 应用程序结构	270
8.2.1	ASP.NET 应用程序元素	270
8.2.2	ASP.NET 应用程序目录结构	271
8.2.3	ASP.NET 的文件类型	271

8.3	Visual Studio	273
8.3.1	Visual Studio 集成开发环境	273
8.3.2	网站和 Web 项目	276
8.3.3	创建无项目文件的网站	276
8.3.4	设计网页	277
8.3.5	编码模型	278
8.3.6	Web 项目	281
8.4	本章小结	283
8.5	习题	283
第 9 章	ASP.NET 常用控件与 Page 类	284
9.1	Web 窗体	284
9.1.1	HTML 标记	285
9.1.2	HTML 控件	287
9.1.3	服务器控件标记	288
9.2	ASP.NET 常用服务器控件	288
9.2.1	服务器控件的类型	288
9.2.2	所有控件的共同属性	289
9.2.3	标准服务器端控件	290
9.2.4	输入验证控件	296
9.2.5	用户控件	301
9.3	ASP.NET Page 类	305
9.3.1	Page 对象	305
9.3.2	Response 对象和 Request 对象	309
9.3.3	Server 对象	313
9.3.4	Session 对象	315
9.3.5	Application 对象	317
9.3.6	Cookie 对象	319
9.4	本章小结	320
9.5	习题	321
第 10 章	数据访问	323
10.1	ADO.NET 基础	325
10.2	基本 ADO.NET 类与存储过程	326
10.2.1	Connection 类	326
10.2.2	Command 类和 DataReader 类	328
10.2.3	DataSet 类	331
10.2.4	DataAdapter 类	333
10.2.5	存储过程	337
10.3	数据绑定	341

10.3.1	单值绑定	341
10.3.2	重复值绑定	343
10.3.3	数据源控件	345
10.3.4	联合使用数据源和数据绑定控件	346
10.4	本章小结	351
10.5	习题	352
第 11 章	案例——网络购物商城	353
11.1	需求分析	353
11.1.1	项目背景	353
11.1.2	需求定义	353
11.1.3	开发环境	354
11.2	软件设计	354
11.2.1	架构设计	355
11.2.2	数据库设计	355
11.2.3	界面设计	358
11.3	功能实现	360
11.3.1	MasterPages 母版页实现	360
11.3.2	主页面实现	363
11.3.3	购物车的实现	367
11.3.4	详细信息实现	368
11.3.5	结算中心的实现	372
11.4	本章小结	374
参考文献		375

第 1 章 绪 论

本章首先对网络编程基础知识做了介绍,重点包括网络应用程序所处的地位,Internet 通信中网间进程的标识等;其次,着重分析了 TCP/IP 协议族中高效的用户数据报协议 (UDP)和可靠的传输控制协议(TCP)及其特点,并以此为基础,介绍了客户机/服务器模型的重要性和工作过程;最后,对目前主要的网络编程技术分类予以阐述。

1.1 网络编程基础

1.1.1 网络应用程序所处的地位

从计算机网络体系结构的角度来看,网络应用程序处于网络层次结构的最上层,如图 1-1 所示。

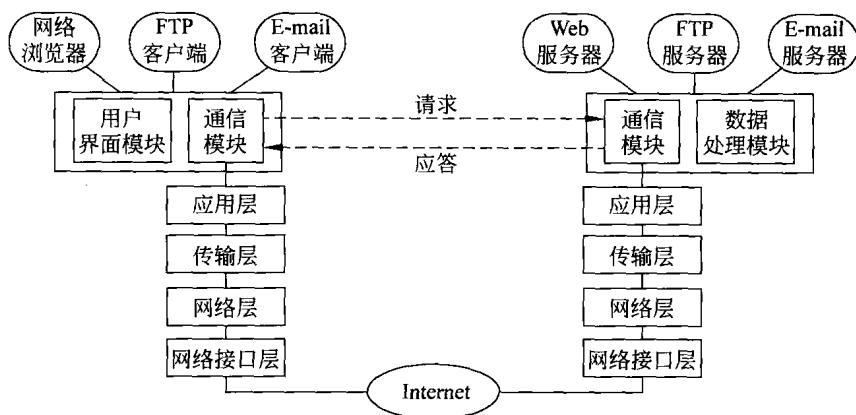


图 1-1 网络应用程序在网络体系结构中的位置

从功能上可以将网络应用程序分为两部分:一部分是专门负责网络通信的模块,它们与网络协议栈相连接,借助网络协议栈提供的服务完成网络上数据信息的交换;另一部分是面向用户或者进行其他处理的模块,它们接收用户的命令,或者对借助网络传输过来的数据进行加工。这两部分模块相互配合,关系密切,缺一不可。首先,通信模块是网络分布式应用的基础,其他模块则对网络交换的数据进行加工处理,从而满足用户的种种需求;其次,网络应用程序最终要实现网络资源的共享,共享的基础就是必须能够通过网络轻松地传递各种信息。

由此可见,网络编程首先要解决网间进程通信的问题,然后才能在通信的基础上开发各种应用功能。

1.1.2 网间进程通信面临的问题

网间进程通信不同于单机进程通信。就单机进程而言,每个进程都有自己独立的地址

空间,同时,操作系统为单机进程通信提供了形式多样的手段,包括管道(Pipe)、软中断信号(Signal)、消息(Message)、共享存储区(Shared Memory)以及信号量(Semaphore)等,从而确保单机进程间的通信既不互相干扰,又能协调一致。然而,网间进程通信是指网络不同主机上的应用进程之间的相互通信,上述通信方式只适合单机进程通信,无法应用到网间进程通信中,因此,网间进程通信面临着新的问题和挑战,主要体现在以下几点。

1. 网间进程的标识问题

在同一主机中,不同的进程可以用进程号(Process ID)唯一标识。但在网络环境下,网间进程可能拥有相同的进程号,例如,主机 A 中某进程的进程号是 5,在 B 主机中也可以存在 5 号进程,此时仅仅说“5 号进程”间的通信就没有意义了,显然,各主机独立分配的进程号已经不能唯一地标识一个进程。

2. 与网络协议栈连接的问题

网间进程的通信离不开网络协议栈的支持。应用进程把数据交给下层的传输层协议实体,调用传输层提供的传输服务,传输层及其下层协议将数据层层向下递交,最后由物理层将数据变为信号,发送到网上,经过各种网络设备的寻径和存储转发,才能到达目的主机。目的主机的网络协议栈再将数据层层上传,最终将数据送交接收端的应用进程,这个过程是非常复杂的。因此,网间进程通信的实现必须要有一种非常简单的方法,使用户不用考虑底层网络协议栈的工作过程,只要处理好上层应用程序所涉及到的相关问题即可。类似的方法不少,其中最具代表性的是基于套接字的网络编程方法。

3. 多重协议的识别问题

操作系统所支持的网络协议种类繁多,常见的有 TCP/IP、IPX/SPX 等。不同协议的工作方式不同,地址格式也不同,一般情况下,不同协议之间不能进行通信,因此,网间进程通信必须解决多重协议的识别问题。

4. 不同的通信服务的问题

不同的应用对象关于网间进程之间的通信服务会有不同的需求。就拿文件传输服务来说,传输的文件可能很大,要求传输非常可靠,无差错,有序,不丢失。试想,花费很长时间下载一个程序,结果因为丢失几个字节导致该程序无法使用,这是多么令人生气的事啊!但是,对于网上聊天这样的应用,要求就不高。在 TCP/IP 协议族中,在传输层有 TCP 和 UDP 这两个协议,TCP 协议提供可靠的字节流传输服务,UDP 协议提供不可靠的数据报传输服务。因此,要求网络应用程序能够有选择地使用网络协议栈提供的网络通信服务功能。

1.1.3 网间进程的标识

Internet 的基础是 TCP/IP 协议,其中网络层协议作用尤为重大。它用 IP 地址统一了 Internet 中各种主机的物理地址,用 IP 数据报统一了各种物理网的帧,实现了异构网的互连。粗略地说,在 Internet 中,每一台主机都有一个唯一的 IP 地址,利用 IP 地址可以唯一地定位 Internet 中的一台计算机,实现计算机之间的通信。但是最终进行网络通信的不是整个计算机,而是计算机中的某个应用进程。每个主机中有许多应用进程,仅有 IP 地址是无法区分一台主机中多个应用进程的。从这个意义上讲,网络通信的最终地址就不仅仅是主机的 IP 地址了,还必须包括可以描述应用进程的某种标识。

就传输层而言,其与网络层在功能上的最大区别就是提供了端到端的进程通信能力,通

过端口成功地解决了通信进程的标识问题。应用层的多个进程通过各自的端口复用 TCP 或 UDP, TCP 或 UDP 再复用网络层的 IP, 经过通信子网的存储转发, 将数据传送到目的主机。而在目的主机中, IP 将数据分发给 TCP 或 UDP, 再由 TCP 或 UDP 通过特定的端口传送给相应的进程。对于网络协议栈来说, 在发送端是自上而下地复用, 在接收端是自下而上地分用, 从而实现了网络中应用进程之间的通信。

1. 端口

端口是 TCP/IP 协议族中应用层进程与传输层协议实体间的通信接口, 在 OSI 七层协议的描述中, 将它称为应用层进程与传输层协议实体间的服务访问点(SAP)。应用层进程通过系统调用与某个端口进行绑定, 然后就可以通过该端口接收或发送数据, 因为应用进程在通信时, 必须用到一个端口, 它们之间有着一一对应的关系, 所以可以用端口来标识通信的网络应用进程。

端口标识符是一个 16 位的整数, 因此, TCP 和 UDP 都可以提供 65535 个端口供应用层的进程使用。由于传输层拥有两个完全独立的协议, 即 TCP 和 UDP, 因此各自的端口号也相互独立。如 TCP 有一个 255 号端口, UDP 也有一个 255 号端口, 二者并不冲突。

端口是操作系统可分配的一种资源, 是一种抽象的软件机制, 包括一些数据结构和 I/O 缓冲区。应用程序通过系统调用与某端口建立绑定关系后, 传输层传给该端口的数据都被相应进程接收, 相应进程发给传输层的数据都通过该端口输出。在 TCP/IP 的实现中, 端口操作类似于一般的 I/O 操作, 进程获取一个端口, 相当于获取本地唯一的 I/O 文件, 可以用一般的读写原语访问它。

端口号的分配是一个重要问题。当网间进程 A 要向 B 发送信息时, A 必须知道 B 的地址, 包括 IP 地址和端口号。由于 IP 地址是全局分配的, 能保证全网的唯一性; 然而, 端口号是由每台主机自己分配的, 只有本地意义, 无法保证全网唯一, 又该如何分配呢? 实际上, 进行端口分配时, TCP/IP 采用的是全局分配(静态分配)和本地分配(动态分配)相结合的方法, 将 TCP 和 UDP 的所有端口分为保留端口和自由端口两部分。

保留端口又称为周知端口(Well-known Port), 范围是 0~1023, 采用全局分配或集中控制的方式, 由一个权威机构根据需要进行统一分配, 静态地分配给 Internet 上著名的服务器进程。这样, 每一个标准的服务器都拥有了一个全网公认的端口号, 在不同的服务器类主机上, 使用相同应用层协议的服务器的端口号也相同, 例如, 所有的 WWW 服务器默认的端口号都是 80, FTP 服务器默认的端口号都是 21。其余的端口, 1024~65535, 称为自由端口, 采用本地分配或动态分配的方法, 由本地操作系统动态地、自由地分配给要进行网络通信的应用层进程。

具体来说, TCP 和 UDP 端口的分配规则如下。

- (1) 端口 0: 不使用, 或者作为特殊的用途。
- (2) 端口 1~255: 保留给特定的服务, 如众所周知的服务。
- (3) 端口 256~1023: 保留给其他服务, 如路由。
- (4) 端口 1024~4999: 可以用做任意客户的端口。
- (5) 端口 5000~65535: 可以用做用户的服务器端口。

2. 半相关

在 Internet 中, 通信的两个进程分别在不同的计算机上, 这两台计算机可能位于不同的

网络中。这些网络通过网关、网桥和路由器等网络互连设备相互连接。因此,要在 Internet 中定位一个应用进程,需要以下三级寻址:

(1) 某一主机总是与某个网络相连,必须指定主机所在的特定网络的地址,称为网络 ID;

(2) 网络中每一台主机应有其唯一的地址,称为主机 ID;

(3) 每一主机上的每一应用进程应有在该主机上的唯一标识符。

在 TCP/IP 中,主机 IP 地址就是由网络 ID 和主机 ID 组成的,IPv4 中用 32 位整数值表示;应用进程则是用 TCP 或 UDP 的 16 位端口号来标识的。

综上所述,在 Internet 中,用一个三元组可以全局唯一地标识一个网间进程:

网间进程=(传输层协议,主机 IP 地址,端口号)

这样一个三元组,叫做一个半相关(Half-association),它标识了 Internet 中进程间通信的一个端点,也把它称为进程的网络地址。

3. 全相关

在 Internet 中,一个完整的网间通信需要由两个进程组成,这两个进程是通信的两个端点,只能使用同一种传输层协议。也就是说,不可能通信的一端用 TCP,而另一端用 UDP。因此一个完整的网间通信需要一个五元组全局唯一地来标识:

(传输层协议,本地机 IP 地址,本地机端口,远端机 IP 地址,远端机端口)

这个五元组称为一个全相关(Association),即两个协议相同的半相关才能组合成一个合适的全相关,或完全指定一对网间通信的进程,以此标识网络中进程间的通信。

1.2 基本网络协议

网络协议是网络通信的基础,网络软件的开发离不开网络协议的支持。作为网络软件开发人员必须了解常用网络协议的基本特点,掌握网络协议在程序设计中的工作方式和流程。

1.2.1 一般特点

在网络分层体系结构中,各层之间是严格单向依赖的,各层次的分工和协作集中体现在相邻层之间的接口上。“服务”是描述相邻层之间关系的抽象概念,是网络中各层向紧邻上层提供的一组服务。下层是服务的提供者,上层是服务的请求者和使用者。服务的表现形式是原语(Primitive)操作,一般以系统调用或库函数的形式提供。系统调用是操作系统内核向网络应用程序或高层协议提供的服务原语。网络中的 n 层总要向 $n+1$ 层提供比 $n-1$ 层更完备的服务,否则 n 层就没有存在的价值。

在 OSI 的术语中,网络层及其以下各层又称为通信子网,只提供“点对点”的通信,没有程序或进程的概念;而传输层实现的是“端到端”的通信,引进了网间进程通信的概念,同时也要解决差错控制、流量控制、报文排序和连接管理等问题。为此,传输层采用两种不同的协议向应用层提供不同的服务:一种是用户数据报协议(User Datagram Protocol,UDP),这是一种面向消息、无连接的、不可靠但效率很高的协议;另一种是传输控制协议(Transfer Control Protocol,TCP),是一种基于字节流的、面向连接的、可靠的协议。网络编程时,要